

MetaProg*
Quality in Software

From C(VS) to shining C(D)
Developing Software the eXtreme Way

Joseph Pelrine
C*O, MetaProg GmbH
jpelrine@metaprog.com

* Copyright © 2002 MetaProg GmbH

MetaProg*
Quality in Software

If you don't live on the edge, you're taking up
too much space

-Doug Robinson

* Copyright © 2002 MetaProg GmbH

MetaProg*
Quality in Software

<refrain>

* Copyright © 2002 MetaProg GmbH

MetaProg*
Quality in Software

Package early, package often

* Copyright © 2002 MetaProg GmbH

MetaProg*
Quality in Software

Automate all that you can

* Copyright © 2002 MetaProg GmbH

MetaProg*
Quality in Software

Take advantage of the tools

* Copyright © 2002 MetaProg GmbH

MetaProg*
Quality in Software

Remember how it all began

* Copyright © 2002 MetaProg GmbH

MetaProg*
Quality in Software

</refrain>

* Copyright © 2002 MetaProg GmbH

MetaProg*
Quality in Software

Inside every large system there's a small system trying to get out

- Chet Hendrickson

* Copyright © 2002 MetaProg GmbH

MetaProg*
Quality in Software

Context

Size	Successful	Challenged	Impaired
Under \$500K	38%	44%	19%
\$501K - \$3M	27%	52%	21%
\$3M - \$6M	16%	55%	29%
\$6M - \$10M	4%	57%	39%
Over \$10M	0%	66%	34%

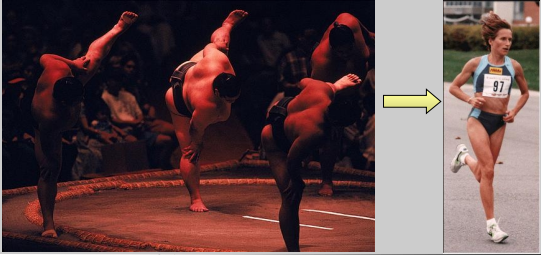
Source - The Standish Group CHAOS'2000 Survey

► This is what Agile Methodologies are trying to address

* Copyright © 2002 MetaProg GmbH

MetaProg*
Quality in Software

Lighten Up! *

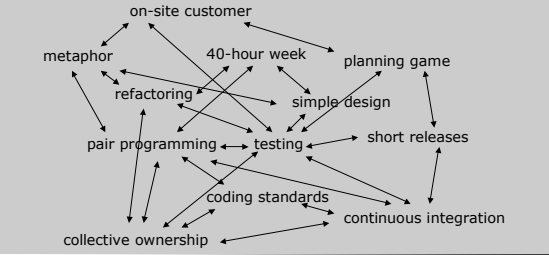


* Gratuitous use of clip-art

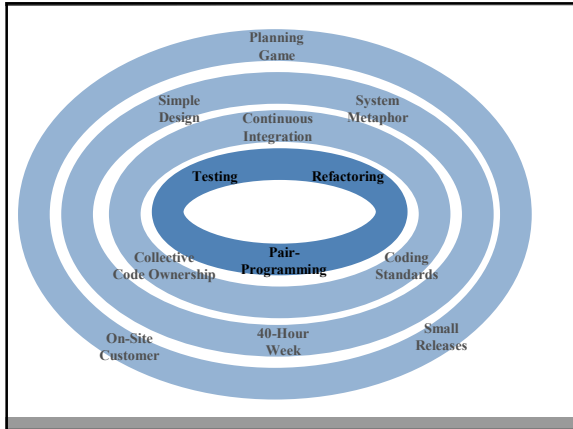
* Copyright © 2002 MetaProg GmbH

MetaProg*
Quality in Software

Bringing order to chaos



* Copyright © 2002 MetaProg GmbH



MetaProg*
Quality in Software

The Coding Circle

- ▶ Testing
- ▶ Refactoring
- ▶ Pair-Programming

* Copyright © 2002 MetaProg GmbH

MetaProg*
Quality in Software

The Automation Circles - Peeling the Onion

- ▶ Compiling (command-line or IDE)
- ▶ Batch or Shell Script
- ▶ Ant
- ▶ Cruise Control

* Copyright © 2002 MetaProg GmbH

MetaProg*
Quality in Software

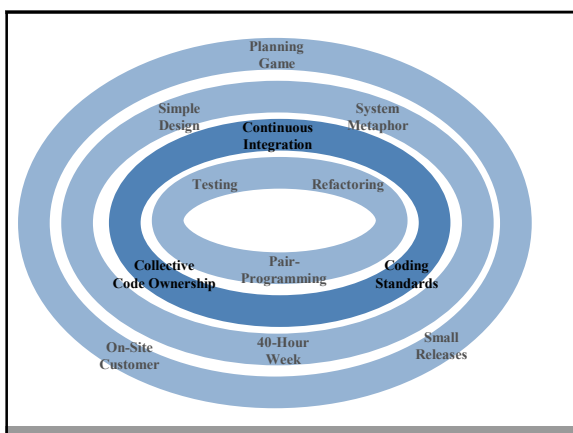
Package early, package often

Automate all that you can

Take advantage of the tools

Remember how it all began

* Copyright © 2002 MetaProg GmbH



MetaProg*
Quality in Software

The Team Circle

- ▶ Coding Standards
- ▶ Collective Code Ownership
- ▶ Continuous Integration

* Copyright © 2002 MetaProg GmbH

MetaProg*
Quality in Software

What is Agile Delivery ?

- ▶ A trendy word to catch your attention !
- ▶ Better known as Continuous Integration
- ▶ Definition 1 : "Ability to have a fully working system at any time"
- ▶ Definition 2 :

```

    graph LR
      A[Development Platform] -- All the time --> B[Pre-production Platform]
    
```

- ▶ Definition 3 : Demo !

* Copyright © 2002 MetaProg GmbH

```

Change detected █
Compiling ... █
Checking coding standard compliance ... █
Running logic unit tests ... █
Generating javadoc ... █
Generating web site documentation ... █
Packaging and deploying application ... █
Running integration unit tests ... █
Running functional unit tests ... █
Running functional system tests ... █
Running performance and load tests ... █
Gathering metrics ... █
Publishing build results to intranet ... █
Build completed in 5 seconds █
System fully operational. Sleeping for 10 min. █
    
```

MetaProg*
Quality in Software

Italians are good at driving trains, they're just bad at making schedules

-David Parnas

* Copyright © 2002 MetaProg GmbH

MetaProg*
Quality in Software

Why do deployment problems happen? 1/2

- ▶ Traditionally, it happens in 2 steps :

- ▶ But ...
- ▶ ... The second step never works if performed at the end.

* Copyright © 2002 MetaProg GmbH

MetaProg*
Quality in Software

Why ? 2/2

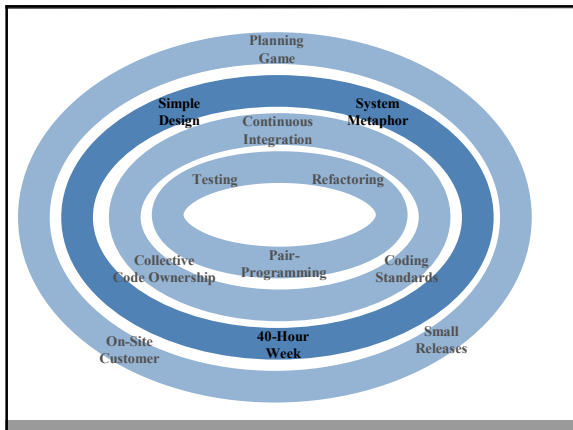
- ▶ Because ...
- ▶ Always underestimated !
- ▶ Only half the work is done ...
 - ▶ Staff training
 - ▶ Bugs cropping up
 - ▶ Realise (Hardware | Architecture | Implementation | Performance | Administrability | etc) is not ok
- ▶ Not what the customer wants syndrome ©
- ▶ Money lenders/Customers getting tired of not seeing anything
- ▶ Team losing steam and focus
- ▶ Cost of integration is becoming higher and higher
 - ▶ Infrastructure tools are improving, by covering more technical and business aspects
 - ▶ => More configuration
 - ▶ Virtual companies with several interconnected IS
 - ▶ => More IS integration

* Copyright © 2002 MetaProg GmbH

MetaProg*
Quality in Software

Package early, package often
Automate all that you can
Take advantage of the tools
Remember how it all began

* Copyright © 2002 MetaProg GmbH



MetaProg*
Quality in Software

The Process Circle

- ▶ Simple Design
- ▶ System Metaphor
- ▶ 40-Hour Week

* Copyright © 2002 MetaProg GmbH

MetaProg*
Quality in Software

Why you need Automation

- ▶ Prevent mistakes
- ▶ Reduce tedium
 - ▶ "Pragmatic programmers are lazy"
- ▶ Increase productivity
- ▶ Capture knowledge
 - ▶ Raise the truck number
- ▶ Maintain security
- ▶ Apply or enforce standards

* Copyright © 2002 MetaProg GmbH

MetaProg*
Quality in Software

What is Automation - approach

- ▶ Ad hoc
 - ▶ "Hail the power of the command line"
 - ▶ IDE macros, shell / batch scripts, cron jobs
 - ▶ Think: refactor mercilessly, avoid spaghetti
- ▶ Integrated
 - ▶ Suite or framework of events, actions, and information flow
 - ▶ Think: YAGNI

* Copyright © 2002 MetaProg GmbH

MetaProg*
Quality in Software

What can be automated?

- ▶ All traditional project phases
 - ▶ Design
 - ▶ Code
 - ▶ Compile
 - ▶ Test
 - ▶ Release
- ▶ Minimal "continuous integration" in XP is the automation of compile and test

* Copyright © 2002 MetaProg GmbH

MetaProg*
Quality in Software

How to automate

<ul style="list-style-type: none"> ▶ Activity ▶ Generate ▶ Build ▶ Exercise ▶ Do ▶ Inspect ▶ Alert 	<ul style="list-style-type: none"> ▶ Tool ▶ XSL, Anakia, PHP, <modelling tool>, ... ▶ Make, Ant ▶ XUnit (and many extensions) ▶ Batch scripts - cmd, sh, expect ▶ Embedded scripting - BSF, VBA ▶ grep, tail, <metrics tool> ▶ Email, web page, SNMP, syslog, ...
---	---

* Copyright © 2002 MetaProg GmbH

MetaProg*
Quality in Software

When to Automate

- ▶ Start of a project
- ▶ As team or scope grows
- ▶ When duplication becomes repetition
 - ▶ "Three strikes and you automate"
- ▶ All the time
 - ▶ Automation is refactoring for the project

* Copyright © 2002 MetaProg GmbH

MetaProg*
Quality in Software

When not to Automate

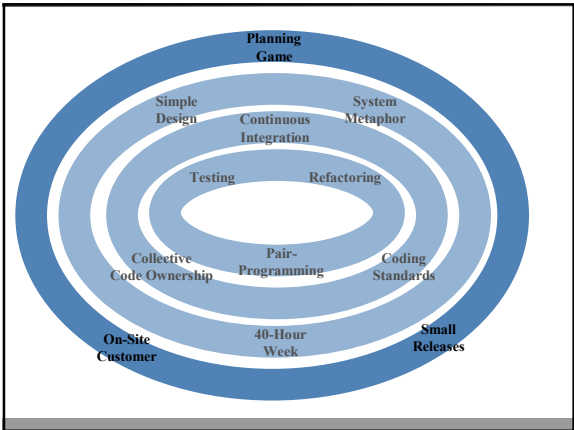
- ▶ Start of a project!
 - ▶ Predicting the future is futile
- ▶ When there's a simpler thing that could possibly work
 - ▶ That you haven't done manually n times before

* Copyright © 2002 MetaProg GmbH

MetaProg*
Quality in Software

Package early, package often
Automate all that you can
Take advantage of the tools
Remember how it all began

* Copyright © 2002 MetaProg GmbH

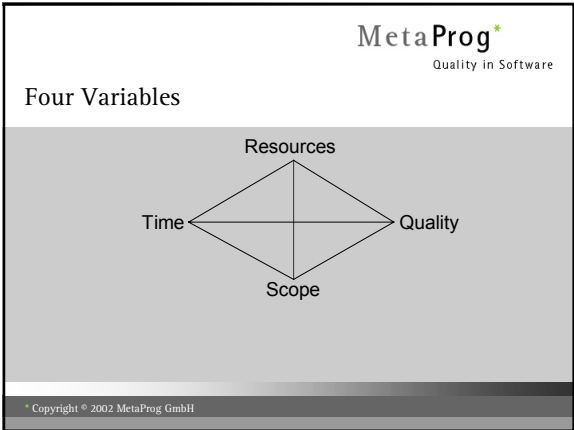


MetaProg*
Quality in Software

The Product Circle

- ▶ Planning Game
- ▶ Small Releases
- ▶ On-site Customer

* Copyright © 2002 MetaProg GmbH



MetaProg*
Quality in Software

Package early, package often

Automate all that you can

Take advantage of the tools

Remember how it all began

* Copyright © 2002 MetaProg GmbH

MetaProg*
Quality in Software

The Four Circles

- ▶ **Coding Circle**
- ▶ **Team Circle**
- ▶ **Process Circle**
- ▶ **Product Circle**

* Copyright © 2002 MetaProg GmbH

MetaProg*
Quality in Software

The Automation Circles - Peeling the Onion

- ▶ **Compiling**
- ▶ **Batch or Shell Script**
- ▶ **Ant**
- ▶ **Cruise Control**

* Copyright © 2002 MetaProg GmbH

MetaProg*
Quality in Software

Summary

- ▶ **Automation is your friend**
 - ▶ It benefits you and your project
- ▶ **There are a lots of tools that help you**
 - ▶ But you won't need them all
- ▶ **Automation fits the agile paradigm**
 - ▶ Apply it first where you need it most
 - ▶ Adopt an iterative and incremental approach

* Copyright © 2002 MetaProg GmbH

MetaProg*
Quality in Software

Resources

- ▶ **Ant and Alexandria**
 - ▶ <http://jakarta.apache.org/>
- ▶ **Cruise Control**
 - ▶ <http://cruisecontrol.sourceforge.net/>
- ▶ **xUnit and extensions**
 - ▶ <http://www.xprogramming.com/>
- ▶ **"The pragmatic programmer"**
 - ▶ ISBN0-201-61622X
- ▶ **"Continuous integration"**
 - ▶ <http://www.martinfowler.com/>

* Copyright © 2002 MetaProg GmbH

MetaProg*
Quality in Software

Thanks

- ▶ Slides available at www.metaprogram.com
- ▶ **Upcoming book: Bitter XP (with Laurent Bossavit)**

* Copyright © 2002 MetaProg GmbH